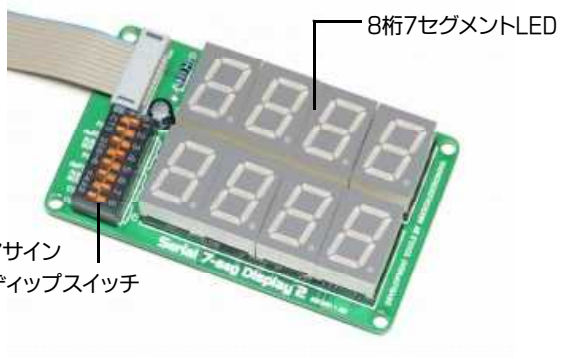
型番:7SEG-DISP2　　REV.1.00

SPI通信式4桁2行7セグメントLEDユニット

取扱説明書
お使いになる前にこの説明書をよくお読みの上正しくお使いください。

(C)2014 マイクロテクニカ

モジュール本体



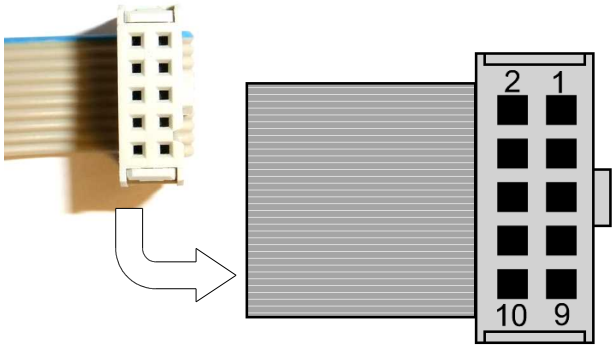
パッケージの内容

■同梱物
・7SEG-DISP2本体

フラットケーブルのピンアサインとディップスイッチ

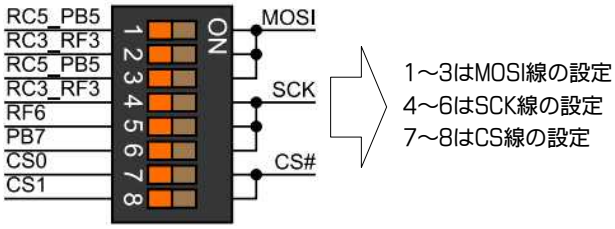
7SEG-GISP2には、本体から10ピンのフラットケーブルが延びています。ケーブルの先には10ピンヘッダピン用のメスコネクタが装着されています。ピンアサインは、基板上のディップスイッチによって切り替えられるようになっています。

ディップスイッチの設定とピンアサインについて下表にまとめます。



フラットケーブル先端のコネクタを下から見た場合のピン番号

ディップスイッチは、データ線のMOSI、SCK、CSの3線を上記コネクタのどのピンに接続するかを設定します。



ディップスイッチの各スイッチをON位置にした時に接続され、コネクタのピン番号との対応は下表の通りです。

	ディップスイッチの番号	コネクタのピン
MOSI線	1	6
	2	4
	3	6
SCK線	4	4
	5	7
	6	8
CS線	7	1
	8	2
Vcc（+5V）		9
GND		10

当方販売中の各種PIC開発統合評価ボードシリーズをお持ちの方は、評価ボードの右側にある10ピンヘッダに直接7SEG-DISP2を接続できます。

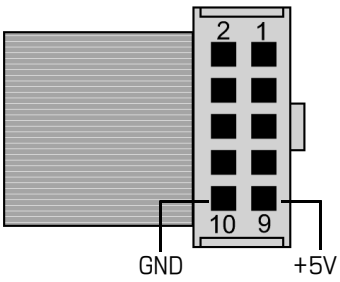
電源の接続と、インターフェイスピンの接続

■電源電圧の接続

7SEG-DISP2の電源電圧は+5Vです。許容範囲は±5%です。リプルやノイズが重畳していない品質のよい電源をご使用ください。

消費電流は、LED無点灯時が約50μA、LED全点灯時で約250mA程度です。余裕を持った電源をご使用ください。

電源ピンは、フラットケーブル先端のメスコネクタの9ピンと10ピンです。9ピンがVcc(+5V)、10ピンがGNDです。



無点灯時から全点灯時に以降する際、いっきに電流が流れますので、瞬間的な電圧降下や他の回路への影響を抑える目的で、VccピンとGND間に100μA程度の電解コンデンサを入れると、回路網への影響を低減できます。

■SPI通信線の接続

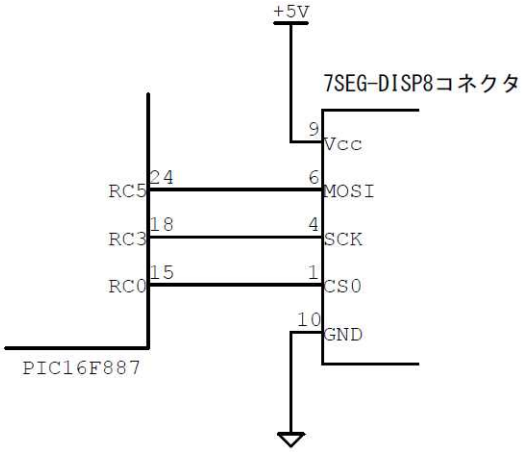
SPI通信はクロック同期式の通信で、データはクロック信号に同期させながら送信します。受信側はクロック信号の立ち上がりで信号を読みます。

7SEG-DISP2では、7SEG-DISP2側（スレーブ側）からマスター側

(マイコン等)へのデータ送信は行いませんので、ホスト側からスレーブ側(7SEG-DISP2側)へデータを送信するだけです。よって、MOSI線(Mater Out Slave In)と、クロックのSCK線、チップセレクトのCS線の3線で通信を行います。

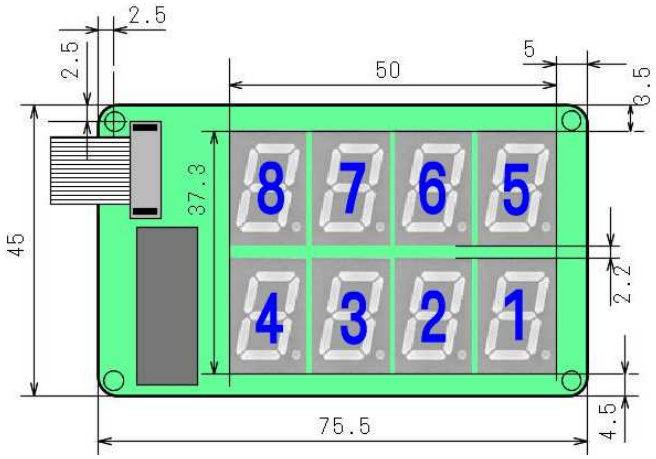
ロジック信号の電圧レベルは、TTLレベルで0V-5Vの5Vp-pです。5V系マイコンなどと直結して使用できます。

ハードウエアUSART機能搭載のPICマイコンと接続する場合には、PICマイコンのSDOピンにMOSI線、SCKピンにSCK線を接続します。CS線は、ソフトウエアで処理しますので適当なピンと接続します。下記にPIC16F887と接続した場合の参考例を記載しました。



上図は、7SEG-DISP2のディップスイッチ、1番・4番・7番をON位置に設定した場合の例です。

7セグディスプレイの桁指定番号

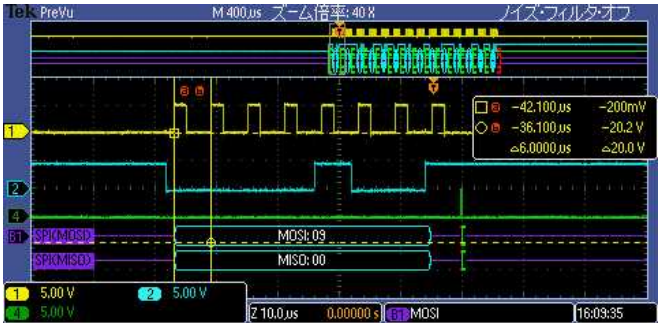


7SEG-DISP2の桁指定は上図の通りとなっています。例えば上の4桁を使う場合には千の位が8番、一の位が5番となります。SPI通信で桁数指定する時には上記の番号を使います。

SPIインターフェイスについて

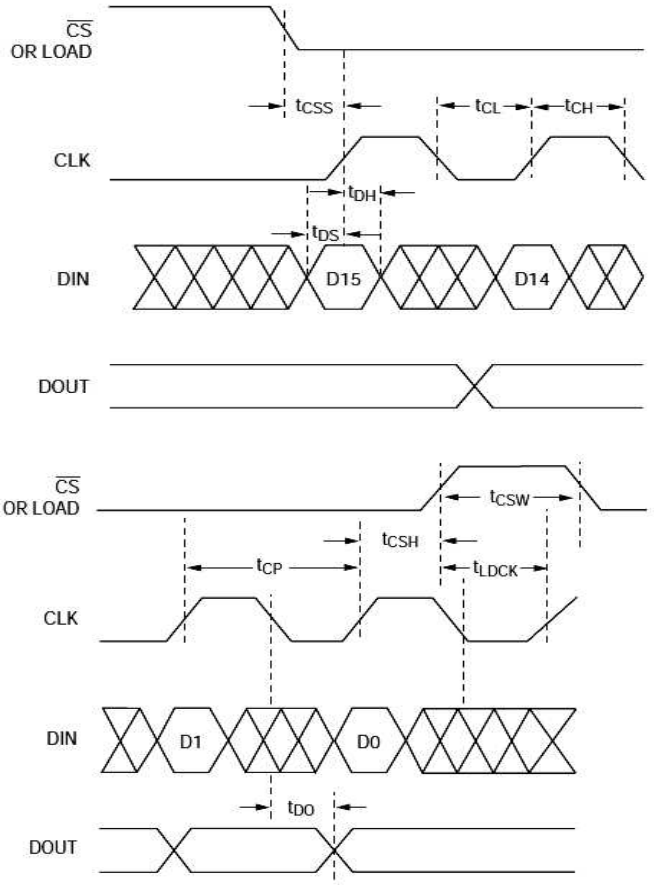
SPI通信は本来、MOSI・MISO・SCL・SS(CS)の4線でクロック同期式の通信を行う方法です。MOSIはマスターアウトスレーブイン、MISOはマスターインスレーブアウトの略です。

SPI通信では常にマスター側のデバイスがクロックを発生させて、そのクロックに同期して通信を行います。7SEG-DISP2では、マスターからデータやコマンドを出力して、スレーブすなわち7SEG-DISP2が受信するという片方向の通信ですので、MOSIとSCL、CSの3線で通信を行うことになります。下図はマスター機器(PICマイコン等)から0x09を7SEG-DISP2側に送信した場合の波形を観察したものです。



上記の例では、クロックは1サイクル6µsなので約166.6KHzのクロック周波数となっています。

SPI通信は、クロック同期式なので厳密な通信速度の管理は必要ありません。ただしクロック幅や立ち上がり時間、ラッチアップ時間は最小時間が決められています。タイミングチャートを下記に示します。



タイミング特性			最小	最大	
CLK Clock Period	t_{CP}		100		ns
CLK Pulse Width High	t_{CH}		50		ns
CLK Pulse Width Low	t_{CL}		50		ns
$\overline{CS}$ Fall to SCLK Rise Setup Time (MAX7221 only)	t_{CSS}		25		ns
CLK Rise to $\overline{CS}$ or LOAD Rise Hold Time	t_{CSH}		0		ns
DIN Setup Time	t_{DS}		25		ns
DIN Hold Time	t_{DH}		0		ns
Output Data Propagation Delay	t_{DO}	C_{LOAD} = 50pF		25	ns
Load-Rising Edge to Next Clock Rising Edge (MAX7219 only)	t_{LDCK}		50		ns
Minimum $\overline{CS}$ or LOAD Pulse High	t_{CSW}		50		ns
Data-to-Segment Delay	t_{DSPD}			2.25	ms

クロック周波数は最小値が100nsですので、最高10MHzまで対応しているということになります。最大値はありませんのでスピードが遅い分には問題になりません。

そのほかは一般的なSPI通信の仕様に準じます。なお、7SEG-DISP2のシリアルデータのフォーマットは16ビットですので、8ビットのデータ列を続けて送信します。(一般的な開発言語はデータ長は8ビットですので、続けて2バイトデータを送信します。)

データの送り方としては、CS信号線がLレベルになって25ns以上時間が経過してから、クロック信号の立ち上がりエッジでデータ信号の状態を読み込みます。データを送信し終えたら、CS信号線をHにしますが、その時の立ち上がりでデータはラッチアップされます。よって、CS線は16ビット目のクロック信号の後にHにする必要があります。

データはMSB先頭で、D15から順番に受信されます。

データフォーマット

■データフォーマット

7SEG-DISP2のデータフォーマットは16ビット長です。上位4ビットは使用しませんので0000とします。下位8ビットはデータです。

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	ADDRESS				MSB			DATA				LSB

データはMSB先頭で、D15から順番に受信されます。

レジスタマップとデータ

■レジスタマップ

レジスタは、12個あり、アドレス1～8までは、7セグメントLEDの各桁を制御するためのレジスタとなっています。9～12までは、輝度調整やデコードモードの設定など、本体の動作に関する設定を行うレジスタとなっています。

16ビット長のデータフォーマットで、上位8ビットでアドレスを指定し、下位8ビットでそのアドレスにセットする値を送ります。

下表はレジスタアドレスマップです。

レジスタ	アドレス					16進コード
	D15～D12	D11	D10	D9	D8	
ディジット0(1桁目)	×	0	0	0	1	0x1
ディジット1(2桁目)	×	0	0	1	0	0x2
ディジット2(3桁目)	×	0	0	1	1	0x3
ディジット3(4桁目)	×	0	1	0	0	0x4
ディジット4(5桁目)	×	0	1	0	1	0x5
ディジット5(6桁目)	×	0	1	1	0	0x6
ディジット6(7桁目)	×	0	1	1	1	0x7
ディジット7(8桁目)	×	1	0	0	0	0x8
デコードモード	×	1	0	0	1	0x9
輝度調整	×	1	0	1	0	0xA
表示桁数指定	×	1	0	1	1	0xB
ディスプレイ表示	×	1	1	0	0	0xC

①桁制御レジスタ(0x01～0x08)

桁制御レジスタは、8桁ある7セグメントLEDの各桁にどんな数字又はセグメント情報を表示するかを指定するためのレジスタです。桁番号は本書2ページの桁指定番号の項目をご覧ください。

②のデコードモードレジスタでデコードモードが有効になっていると、該当する桁レジスタに入った値が直接数字として7セグメントLEDに出力されます。下表はレジスタデーター覧です。

7-SEGMENT CHARACTER	REGISTER DATA					
	D7*	D6–D4	D3	D2	D1	D0
0		X	0	0	0	0
1		X	0	0	0	1
2		X	0	0	1	0
3		X	0	0	1	1
4		X	0	1	0	0
5		X	0	1	0	1
6		X	0	1	1	0
7		X	0	1	1	1
8		X	1	0	0	0
9		X	1	0	0	1
—		X	1	0	1	0
E		X	1	0	1	1
H		X	1	1	0	0
L		X	1	1	0	1
P		X	1	1	1	0
blank		X	1	1	1	1

例えば7セグメントLEDの3番に"5"を表示しようとした場合には、下記のようになります。

0000**0011**0000**0101**
3桁目　　数値(5)

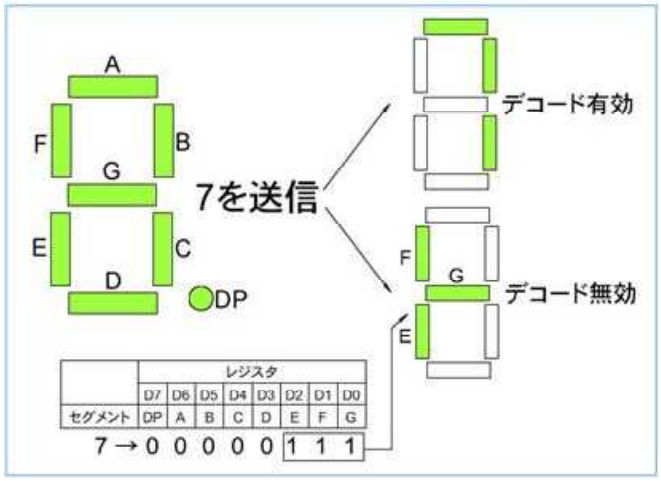
例えばCコンパイラでSPI通信にて値を送信する場合には上位1バイトと下位1バイトを分けて上位から送信します。

```
PORTC.CHIP_SELECT = 0;
SPI_write(3);
SPI_write(5);
PORTC.CHIP_SELECT = 1;
```

同様の方法で1桁目～8桁目までを操作することができます。
なおデコードモードとノンデコードモードでは、7セグメントLEDのセグメント駆動の仕方が違いますので詳しくは、次の②をご覧ください。

②デコードモード

7SEG-DISP2のセグメント駆動には、デコードモードとノンデコードモードがあります。送信した値を「数字として」表示したい場合には、デコードモードを、送信した値を「セグメントデータ」として扱いいた場合にはをノンデコードモードといいます。下図に例を示します。



デコードモードを有効にした場合、"7"という値を送信すると、数字の7を7セグメントLEDに表示します。ノンデコードモードにした場合、"7"という値を送信すると、7は2進表記で00000111ですので、この1に対応する部分のLEDが点灯します。上図ではその例を紹介しています。ノンデコードモード時の各セグメントとレジスタは下表の通りとなっています。

	レジスタ							
	D7	D6	D5	D4	D3	D2	D1	D0
セグメント	DP	A	B	C	D	E	F	G

デコードモードレジスタの設定値は下表の通りです。

デコードモード	レジスタデータ								16進
	D7	D6	D5	D4	D3	D2	D1	D0	
すべての桁で デコードしない	0	0	0	0	0	0	0	0	0x0
1桁目だけデコードし 他桁はデコードしない	0	0	0	0	0	0	0	1	0x1
1桁～4桁はデコードし 他桁はデコードしない	0	0	0	0	1	1	1	1	0xF
全桁デコードする	F	F	F	F	F	F	F	F	0xFF

全桁デコードモードにする場合には、レジスタ9番地に0xFFをセットすることになります。

③輝度設定

7セグメントLEDの輝度を16段階で設定します。設定は0x0～0xFまでで、0x0が最も暗く、0xFが最も明るくなります。(0x0は消灯ではありません。)

④表示桁数指定

7セグメントLEDをONにする桁数を設定します。次の表の通り値を設定します。

表示桁数設定	レジスタデータ								16進
	D7	D6	D5	D4	D3	D2	D1	D0	
1桁目だけ表示	x	x	x	x	x	0	0	0	0x0
1･2桁目表示	x	x	x	x	x	0	0	1	0x1
1～3桁目表示	x	x	x	x	x	0	1	0	0x2
1～4桁目表示	x	x	x	x	x	0	1	1	0x3
1～5桁目表示	x	x	x	x	x	1	0	0	0x4
1～6桁目表示	x	x	x	x	x	1	0	1	0x5
1～7桁目表示	x	x	x	x	x	1	1	0	0x6
全桁表示	x	x	x	x	x	1	1	1	0x7

※　x印は値を問いません

なお、表示桁数は輝度に影響を与えますので、ノンデコードモードに設定している場合には、全レジスタに0を設定して表示をしないようにしている桁に対して、本表示桁数指定の設定でその桁を表示するように設定しないでください。

⑤ディスプレイ表示

全桁の7セグメントLEDの表示をするか、しないかを設定します。レジスタの値が0で表示しない、1で表示する、に設定されます。

実際の使用プログラム例

実際に使用する場合には、先に「デコードモード」「輝度設定」「表示桁数設定」「ディスプレイ表示設定」を行って、その後に各桁に対して表示したい値をセットするようにします。

下記は当方販売中のmikroCとPicBasic Pro Compilerを使用して、7SEG-DISP2を制御している例です。使用しているマイコンはPIC16F887として、CSピンはPIC16F887のRC0に接続されているものとします。表示として"12345678"と表示させます。

■mikroC for PICを使用した場合の例

```
#define CHIP_SELECT F0
unsigned long cnt;

void system_init() {
  PORTC.CHIP_SELECT = 0;
  SPI_write(0x09);
  SPI_write(0xFF);
  PORTC.CHIP_SELECT = 1;

  PORTC.CHIP_SELECT = 0;
  SPI_write(0x0A);
  SPI_write(0xFF);
  PORTC.CHIP_SELECT = 1;

  PORTC.CHIP_SELECT = 0;
  SPI_write(0x0B);
  SPI_write(0x07);
  PORTC.CHIP_SELECT = 1;

  PORTC.CHIP_SELECT = 0;
  SPI_write(0x0C);
  SPI_write(0x01);
  PORTC.CHIP_SELECT = 1;
}

void display_number(unsigned long data) {

  PORTC.CHIP_SELECT = 0;
  SPI_write(8);
  SPI_write(data/10000000);
  PORTC.CHIP_SELECT = 1;

  PORTC.CHIP_SELECT = 0;
  SPI_write(7);
  SPI_write((data/1000000)%10);
  PORTC.CHIP_SELECT = 1;

  PORTC.CHIP_SELECT = 0;
  SPI_write(6);
  SPI_write((data/100000)%10);
  PORTC.CHIP_SELECT = 1;

  PORTC.CHIP_SELECT = 0;
  SPI_write(5);
  SPI_write((data/10000)%10);
  PORTC.CHIP_SELECT = 1;
```

↓続きます

```
  PORTC.CHIP_SELECT = 0;
  SPI_write(4);
  SPI_write((data/1000)%10);
  PORTC.CHIP_SELECT = 1;

  PORTC.CHIP_SELECT = 0;
  SPI_write(3);
  SPI_write((data/100)%10);
  PORTC.CHIP_SELECT = 1;

  PORTC.CHIP_SELECT = 0;
  SPI_write(2);
  SPI_write((data/10)%10);
  PORTC.CHIP_SELECT = 1;

  PORTC.CHIP_SELECT = 0;
  SPI_write(1);
  SPI_write(data%10);
  PORTC.CHIP_SELECT = 1;
}

void main() {

  TRISC.CHIP_SELECT = 0;
  SPI_init();
  system_init();
  display_number(12345678);

}
```

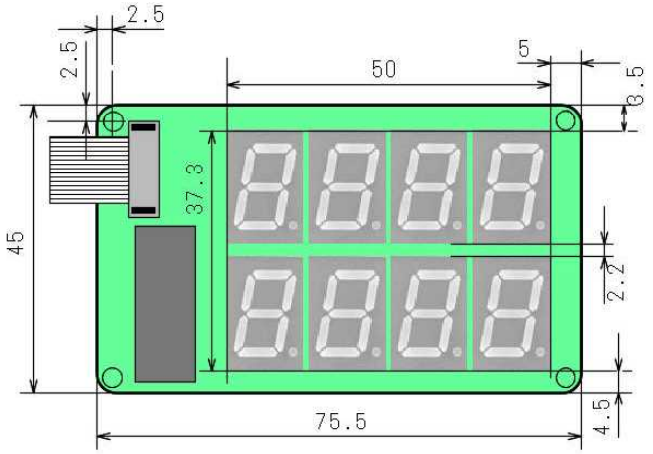
■PicBasic Pro Compilerを使用した場合の例

```
alias: ss var portc.0
       scl var portc.3
       sdo var portc.5

vars:  i var byte
       n var byte

init:  TRISC = 0
       ss = 0
       n = 8

main:
       Gosub sys_init

       For i = 1 to 8
           ss = 0
           SHIFTOUT sdo,scl,1,[i]
           SHIFTOUT sdo,scl,1,[n]
           ss = 1
           n = n - 1
       Next i
       End

sys_init:
       ss = 0
       SHIFTOUT sdo,scl,1,[$09]
       SHIFTOUT sdo,scl,1,[$FF]
       ss = 1

       ss = 0
       SHIFTOUT sdo,scl,1,[$0A]
       SHIFTOUT sdo,scl,1,[$0F]
       ss = 1

       ss = 0
       SHIFTOUT sdo,scl,1,[$0B]
       SHIFTOUT sdo,scl,1,[$07]
       ss = 1

       ss = 0
       SHIFTOUT sdo,scl,1,[$0C]
       SHIFTOUT sdo,scl,1,[$01]
       ss = 1

       return
```

主な仕様

電源電圧:	DC5V（要安定化）
消費電流(平均値):	LED無点灯時:約50uA
	LED全点灯時(全桁8点灯時):約250mA
	12345678点灯時:約176mA
ディスプレイスキャン:	800Hz(typ.)（1.25ms間隔）
ディスプレイドライバ電流:	320mA(min.)
	1μ秒あたり約20mA(typ.)
ロジック電圧Hレベル閾値:	3.5V(min.)
ロジック電圧Lレベル閾値:	0.8V(max.)
ロジックピン電流:	流れ込み電流最小−1μA
	流れ出し電流最大1μA
動作環境:	0℃～70℃ （動作保証範囲）
シリアル通信方式:	同期式シリアル通信 SPI方式
信号電圧レベル:	5Vp−p
生産国:	中国
RoHS指令への対応:	対応済み

外形寸法



（単位:mm）

サポート情報

本製品についてのサポートは電子メールにて受け付けています。なお、サポートでは本製品についてのお問い合わせは承りますが、本製品を駆動するためのプログラム方法やプログラムのデバッグ等は行っておりません。また他社製品についてのお問い合わせについてはお答えいたしかねます。

サポートメールアドレス: support@microtechnica.net

回答までにお時間を頂くことがあります。あらかじめご了承ください。

マイクロテクニカ
〒158-0094 東京都世田谷区玉川1-3-10
TEL: 03-3700-3535 FAX: 03-3700-3548

(C)2014 Microtechnica All rights reserved